

# Unveiling the Test Champions

**Paweł Wieczorek**

**EOSS 2024**

# Agenda

- Motivation
- Candidates
- Use cases
- Trying it out
- Recommendations
- Next steps



COLLABORA

# Motivation

# Farming together

## The vision

A better solution

- We can achieve more if we do it together
  - We can move on from the basics and focus on the stuff that generates value – often testing
  - Reduce the friction when integrating with existing higher layers.
  - At a minimum bring farmers together
- <https://www.youtube.com/watch?v=-oO9y2HnoYs>
  - <https://elinux.org/images/5/52/BoF-FarmingTogether.pdf>



COLLABORA

Open First

# Other efforts

- [https://elinux.org/Automated\\_Testing\\_Summit\\_2018](https://elinux.org/Automated_Testing_Summit_2018)
- [https://elinux.org/Automated\\_Testing\\_Summit\\_2019](https://elinux.org/Automated_Testing_Summit_2019)
  - Test Labs status updates
  - Test stack survey: [https://elinux.org/Test\\_Stack\\_Survey](https://elinux.org/Test_Stack_Survey)
  - Multiple tracks:
    - Board/Lab management
    - Test systems/Test definitions
    - QA systems and issues



# Follow-up actions

- Open Automated Testing Standards VCC:  
[https://elinux.org/Automated\\_Testing#Conference\\_call](https://elinux.org/Automated_Testing#Conference_call)
- Workflows ML:  
<https://lore.kernel.org/workflows/>

# Success?



COLLABORA

Open First

# Test champions

- What could you try out first?
- What else could you out?
- What could be further improved?



# Taking a closer look

- Set up various test labs
- Start small (with virtual devices)
- No external support

# Summary

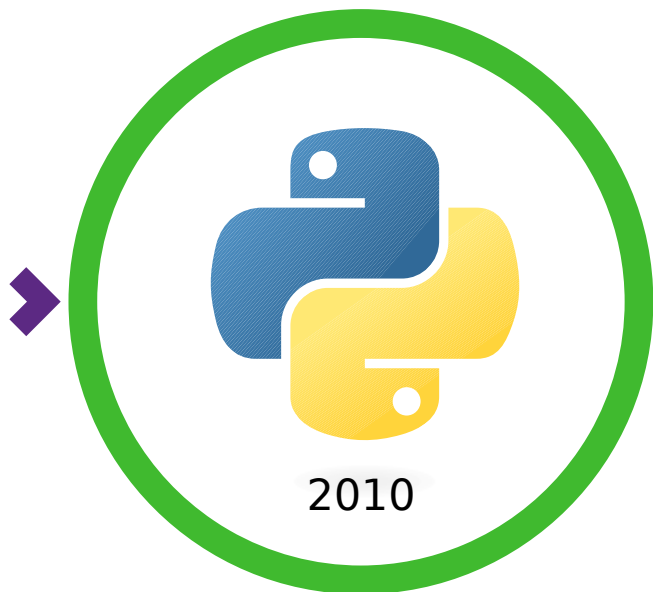
- What are the biggest strengths of discussed solutions?
- Which use cases do they cover best?
- How easy it is to get started on one's own?



COLLABORA

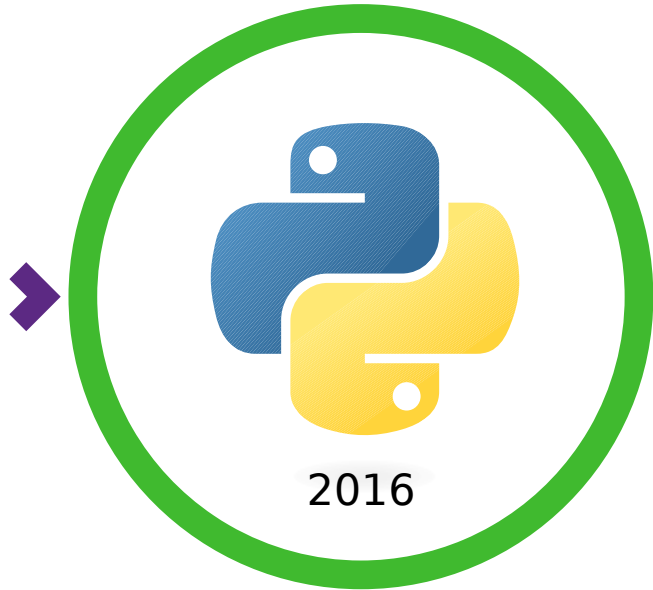
# Candidates

# LAVA



- Django + PostgreSQL
- YAML **test-definitions**
- Dashboard, results processing
- High performance (**tracked**)
- **Cloud-ready**
- Well-integrated

# labgrid



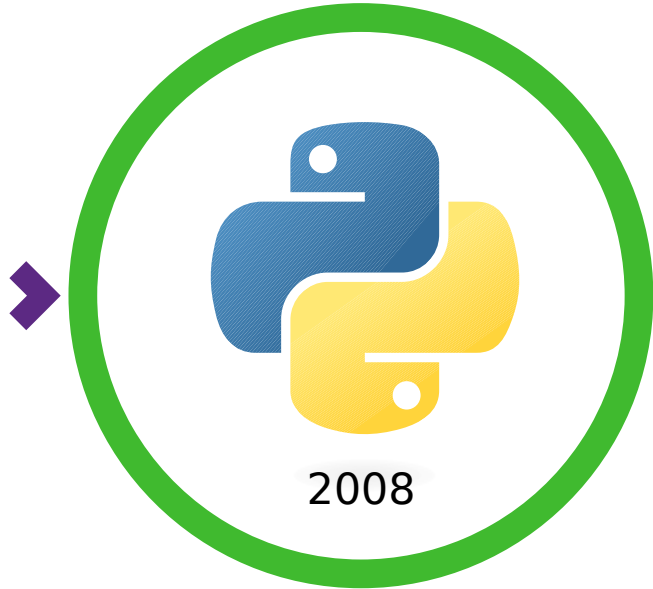
- “Places”
- Impressive HW support
- Interactive sessions
- pytest integration
- Tutorials on YouTube

# boardswarm



- HW abstraction
- Interactive use
- Building block
- Unix philosophy:  
“Does one thing well”

# Beaker



- Enterprise solution
- Similar requirements
- UI + dashboards
- Modular:

Cobbler

# Custom HW-assistance



- SDMux, USBMux, etc.
- Testing In A Box
- Tailored solution, yet modular
- Tricky iterations
- Complex evaluation
- ROI



# Champions

Lab	Start	Language	Test/Task
LAVA	2010	Python	YAML
labgrid	2016	Python	pytest
boardswarm	2022	Rust	N/A
Beaker	2008	Python	XML
Custom HW	2023	N/A	GitLab CI Pipeline





# Use cases

# Interactive access

- Remote devices
- Reservation mechanism
- Additional control/measurements

# Batch processing

- Large number of submissions
- Scheduling + notifications
- Acceptance tests

# Version control systems

- Single-repository/integration projects
- Per-patch verification
- Developer workflow integration

# Documentation and first steps

- Quickstarts
- Examples
- Evaluation environments

# Analyzed aspects

- Interactive access
- Batch processing
- Version control systems
- Documentation and first steps



COLLABORA

# Trying it out



# Interactive access

- labgrid: feature-rich
- boardswarm: keeping it simple
- LAVA: new tmate-based approach (TBE)

# Batch processing

- LAVA:
  - Scheduler
  - Priorities
  - Callbacks
  - Queue cleanup
- External schedulers?

# Version control systems

- Interactive: VCS pipeline becomes regular user
- Scheduled: submit + poll/notify or **use custom runner**
- Built around VCS: configuration kept in a single place

# Documentation

- **Extensive** Beaker docs: setup, admin, user guides
- LAVA docs: [classic](#) and [new](#)
- [labgrid YouTube tutorials](#)

# Examples

- TIAB: examples in most repositories
- labgrid: **examples subdir** in main repository
- LAVA: **test-definitions**

# Evaluation environments

- Beaker: Beaker in a box
- labgrid: Board Farms for Everyone
- LAVA:
  - <https://gitlab.com/lava/pkg/docker-compose>
  - <https://github.com/kernelci/lava-docker>

# What stood out?

- Design decisions: labgrid, boardswarm
- Docs: Beaker, LAVA
- TIAB: possible HW designs reuse



# Recommendations



# No silver bullet

- Ticking all the boxes requires *extreme* luck
- Each champion may require adjustments
- *Some* integration effort involved

# Know your requirements

- Choice instead of a guess
- ROI calculation
- Fewer integration bumps
- Set expectations

# Hybrid approaches

- boardswarm as a base layer replacement
- labgrid with LXA TAC
- Integration solutions:
  - GitLab runner
  - Prometheus metrics exporter

# Engage!

- Don't get stuck and give up
- Reduced time of initial setup
- Feedback to developers



# Next steps

# Evaluate together

- Upstream changes (if applicable)
- Virtual-only quickstart
- GCD environment?

# Evaluation hardware

- Follow-up **previous discussion**
- Slightly more complex
- ROI TBD – would it ever be used?

# Scaling attempts

- Testing a large testing software
- Ghosting the hardware
- Pushing test lab to its limits





COLLABORA

# Summary

# Wrap-up

- Many solutions to choose from
- Up for a challenge?
- Engage!



**Thank you!**



COLLABORA

**Open First**